UNITED STATES PATENT APPLICATION

for

METHOD AND APPARATUS FOR MULTI-CONTACT SCHEDULING

Inventors:

Illah Nourbakhsh
Ofer Matan
Jason Fama
Scott Veach
Edward Hamilton
Alex Fukunaga

Prepared by:
Wilson Sonsini Goodrich & Rosati
650 Page Mill Road
Palo Alto, California 94304-1050

Attorney Docket No.: 20191-704

Express Mail Number : EL757543215US

METHOD AND APPARATUS FOR MULTI-CONTACT SCHEDULING

FIELD OF THE INVENTION

The invention is in the field of generating complex schedules in dynamic environments, such as multi-contact centers.

BACKGROUND

Generating schedules for employees is a complex problem for enterprises. Telephone call center scheduling is an example of a scheduling problem with a large number of variables. Variables include contact volume at a particular time of day, available staff, skills of various staff members, call type (e.g., new order call and customer service call), and number of call queues, where a queue may be assigned a particular call type. A basic goal of call center scheduling is to minimize the cost of operators, or agents, available to answer calls while maximizing service. Quality of service, or service level, can be quantified in various ways. One common metric for call service level is the percentage of incoming calls answered in a predetermined time, e.g. thirty seconds. The call center may receive calls of various types that are assigned to respective call queues.

Traditionally, call center scheduling is performed by first forecasting incoming contact volumes and estimating average talk times for each time period t (based on past history and other measures). The forecast is based upon historical data. Next, a closed-form formula known as reverse Erlang-C is used to compute full-time equivalent (FTE) agent requirement to provide a desired service level for each time period t. Such a method is described in Elementary Queuing Theory and Telephone Traffic, by Petr Beckmann, 1977,

and in Lee's ABC of the Telephone Training Manuals, Geneva, IL. After the FTE agent requirement are computed, the required number of agents is scheduled for each time period t.

At a call center, calls of different types are typically placed onto different queues by an Automatic Call Distributor (ACD). The calls wait at the ACD for an operator to answer them. The ACD is typically for handling telephone calls. Different types of calls are assigned to different call queues. Typically, not all agents have the same skills, and thus some agents can answer some calls while other agents cannot. Scheduling for varying agent skill sets is the skill-based scheduling problem. The skill-based scheduling problem is considerably more difficult than the basic call center scheduling problem because of all the interactions between queues. Typical approaches to solving the skill-based scheduling problem involve variations on an Erlang formula. The Erlang formulas are useful for computing staffing requirements for telephone contacts where the average contact volume is high, service level requirements are stringent, the task of answering a telephone call is not interruptible, and an agent can only answer one telephone call at a given time. Service level is expressed as a percentage of incoming calls that can be answered in within a maximum time limit. An example of stringent service levels is 80%-90% of incoming calls to be answered within 20-60 seconds.

In the past few years, however, call centers have evolved into "contact centers" in which the agent's contact with the customer can be through many contact media. For example, a multi-contact call center may handle telephone, email, web callback, web chat, fax, and voice over internet protocol (IP). Therefore, in addition to variation in the types of calls (e.g., service call, order call), modern contact centers have the complication of variation in contact media. The variation in contact media adds complexity to the agent scheduling process. For example, one of the ways in which contact media can vary markedly is in time

allowed for response to the contact. Telephone calls are typically expected to be answered when they are received, or in "real-time". If a caller does not receive a real-time answer in a fairly short time, the caller hangs up, abandoning the call. If a contact is by email or fax, on the other hand, the customer does not expect a real-time response. Therefore response times for various contact media vary from seconds to days.

Call centers have traditionally had to respond immediately to their telephone customers, and therefore the incoming telephone call queues are called on-line queues. In multi-contact call centers, however, an agent may be required to respond to incoming customer contacts from other queues, such as e-mail and faxed requests, in addition to responding to customer contacts from "immediate" queues, such as telephone calls and computer chats. Email and fax contact do not require immediate responses, but can be deferred. As with traditional telephone call centers, agents can only answer the types of calls for which they have the appropriate training and/or experience. Because all agents must be scheduled across immediate and deferred queues, in addition to all of the traditional scheduling constraints, the multi-contact scheduling problem is considerably complex.

Common techniques for scheduling staff in contact centers that have both immediate and deferred queues are inadequate. For example, in typical scheduling techniques, immediate queues are dealt with in terms of immediate performance measures such as average time to answer and service level. Deferred queues are considered only secondarily. Deferred queues are often simply scheduled into the day during lulls in on-line queue demand. No consideration is given to a projected or expected performance of deferred queues.

There are currently no known methods for effectively computing staffing requirements for e-mail, chat, Web callback, and other new media given certain service level

requirements and contact arrival rates. Erlang formulas cannot be used because off-line contact media do not conform to Erlang's queuing theory models. Some of the aspects of deferred contacts that do not conform with Erlang models include the interruptibility of tasks, the fact that multiple contacts may be handled simultaneously, and the fact that service levels can be in hours or days, rather than seconds. This limits the effectiveness of the multi-contact center because there is no common performance measure for immediate and deferred queues, and thus no way to assess possible trade-offs between assigning agents to immediate queues versus deferred e queues. Another disadvantage of current scheduling methods that a call center manager cannot visualize queue performance in a type-independent manner and therefore must make adjustments to the schedule without the benefit of data to direct the adjustments.

SUMMARY OF THE DISCLOSURE

A method and apparatus for multi-contact scheduling is described. Embodiments of the invention can be used with existing scheduling software to produce agent schedules for contact centers that handle on-line "immediate" and off-line "deferred" contact queues. One embodiment includes scheduling software receiving a scheduling data from a user interface, and the scheduling software generating scheduling constraints. A search engine uses the scheduling constraints to generate potential schedules, including potential schedules for immediate queues, and potential schedules for deferred queues. An analysis is performed on the potential schedules for the immediate queues. The analysis for the immediate queues can be preformed using existing analysis tools. In addition, an analysis is performed on the potential schedules for the deferred queues. The analyses produce estimated service levels expressed in interchangeable units. The immediate and deferred queues can thus be commonly assessed, allowing the choice of a schedule that is optimized both for immediate queues and deferred queues.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of an embodiment of a system for multi-contact schedule generation.

Figure 2 is a simplified flow diagram of an embodiment of schedule generation, including an a schedule analysis adapted to deferred queues.

Figure 3 is a simplified flow diagram of an embodiment of schedule generation for immediate and deferred queues.

Figure 4 is a more detailed flow diagram of an embodiment of a schedule analysis adapted to deferred queues.

DETAILED DESCRIPTION

A method and apparatus for multi-contact scheduling is described. Embodiments of the invention allow scheduling of immediate contact queues and deferred contact queues for a contact center. Potential agent schedules are analyzed and estimated service levels are generated for both immediate queues and deferred queues in common units. An optimized schedule that takes into account all types of contacts can thus be generated.

Figure 1 is an embodiment of a system 100 for generating complex schedules. The system includes multiple client computers 102-105, which are coupled to the server 106 through a network 108. The network 108 can be any network, such as a local area network, a wide area network, or the Internet. The client computers each include one or more processors and one or more storage devices. Each of the client computers also includes a display device, and one or more input devices. The server 106 includes one or more storage devices. All of the storage devices store various data and software programs. In one embodiment, methods for generating complex schedules are carried out on the system 100 by software instructions executing on one or more of the client computers 102-105. The software instructions may be stored on the server 106 or on any one of the client computers. For example, one embodiment is a hosted application used by a call center of an enterprise that requires complex scheduling of many employees, or agents. The software instructions are stored on the server and accessed through the network by a client computer operated by the enterprise. In other embodiments, the software instructions may be stored and executed on the client computer. Data required for the execution of the software instructions can be entered by a user of the client computer through a specialized user interface. Data required for the execution of the software instructions can also be accessed via the network and can be stored anywhere on the network.

One example of a complex schedule is an agent schedule for a multi-contact center, or contact center. A contact center is an organization that responds to incoming contacts from customers of an enterprise. The incoming contacts are via any one of a number of contact media, such as telephone calls, email, fax, web chat, voice over internet protocol, and call backs. An agent is an employee that is trained to respond to various contacts according to both the content and the medium of the contact. Each agent can have a different skill set. For example, one agent may be trained to answer live telephone help inquiries regarding certain products, respond to email regarding certain products, receive telephone purchase orders for certain products, etc. Typically, incoming contacts are assigned to different queues based upon the content and/or medium of the contact. In embodiments of the invention, contact queues are divided into at least two types of queues. One type of queue is an immediate queue for contacts that can be abandoned and should be responded to in real-time, such as telephone calls. Another type of queue is a deferred queue for contacts that cannot be abandoned (at least not immediately) and should be responded to within some time period after receipt, such as email or fax.

An agent may be assigned to multiple contact queues within a time period. A contact queue typically handles one type of contact requiring a particular skill or skills. The possible number of skill sets includes every permutation of combinations of the existing skills in the organization. Each agent has a particular skill set, but the skill sets among different agents may overlap. In embodiments of the invention, as described more fully below, a user who is performing scheduling can produce a schedule that most efficiently uses available agents across contact media, taking into account the widely varying acceptable response times of different contact media. For example, telephone calls must be responded to in seconds, while fax contacts may be responded to in some number of days.

Traditionally there are two important measures for immediate queue performance. One measure is percentage of calls answered, or PCA, which represents the service level provided by the queue. The other measure is based upon the number of agents servicing a queue (agents available) and the number of agents required on the queue (agents required) in order to meet service level goals. Embodiments of the invention estimate values of PCA, agents available, and agents required for deferred queues using discrete mathematical analysis as further described below. Any other performance measures may be used in various embodiments, such as average speed to answer. Embodiments of the invention present the performance measures for immediate and deferred queues in an identical way, which facilitates visualization of potential schedules and human decision-making.

Figure 2 is a simplified flow diagram of an embodiment of schedule generation that is capable of analyzing deferred queue performance and representing that performance using the same measures traditionally used for immediate queues. At 202, a user enters scheduling data via a user interface that is specifically designed for the agent scheduling process. The scheduling data applies to a schedule period that includes multiple intervals of predetermined length. For example, the scheduling period can be one day with intervals of one half-hour. The scheduling data includes the type of contact media, the expected contact volume, the expected contact handle times, service goals, agent designations, and work rules. Some of the data, such as expected contact handle times, is derived from historical data. In one embodiment, the scheduling data includes data for deferred queues. In other embodiments, the scheduling data includes data for immediate and deferred queues.

At 204, scheduling software receives the scheduling data. The scheduling software is an existing tool for analyzing the scheduling data and generating scheduling constraints, including workload forecasts and service goals. The scheduling constraints are sent to a

search engine at 206. The search engine generates potential schedules for analysis. At 208, analysis of schedules for deferred queues is performed to produce estimated service levels for the deferred queues according to the potential schedule that was analyzed. The analysis of 208 is performed using a forward-push discrete event modeler which estimates PCA for deferred queues given the workload and capacity in any given interval within the schedule period. PCA for deferred queues is used by the agent requirement scoring function at 210, along with service goals, to produce an agent requirement score. The analysis of 208 will be described more fully with reference to **Figure 4**. The agent requirement score is used by the search engine 206 to evaluate the schedule. A schedule with the highest agent requirement score of all of the analyzed schedules is output as an "optimal" schedule to the user interface 202. The flow of **Figure 2** produces an optimal schedule, including optimal schedules for deferred queues as measured by traditional metrics used for immediate queues.

In another embodiment, which will now be described with reference to the flow diagram of **Figure 3**, optimal schedules for both immediate and deferred queues are produced in one embodiment, and each type of queue is analyzed separately. One analysis is used for deferred queues, and another analysis is used for immediate queues. At 302, the user enters scheduling data via a user interface. The scheduling data is similar to that described with reference to **Figure 2**. The scheduling data applies to both immediate queues and deferred queues. The scheduling software, at 304, uses the scheduling data to generate scheduling constraints, workload forecast for both immediate queues and deferred queues, and service goals for all queues. At 306, the search engine uses the scheduling constraints to generate potential schedules for both immediate queues and deferred queues. In one embodiment, a single schedule including both types of queues is received by the deferred queue analysis at 308, and by the immediate queue analysis at 310. The deferred queue analysis generates estimated service levels for queues as described with reference to **Figure 2**. The immediate

queue analysis generates estimated service levels according to conventional techniques such as Erlang-based analysis. The estimated service levels for the immediate queues and deferred queues are in the same or interchangeable units, so that both types of queues are scored together by the agent requirement scoring function at 312. This generates a score that reflects the effectiveness of the potential schedule in utilizing all of the available agents, with their varying skill sets, across different contact queues. An agent requirement score is received by the search engine 306, which designates an optimal schedule. In one embodiment, a schedule is evaluated for each queue in the schedule. For example, each queue will have potentially different agents available and agents required. If some of the queues are deferred, and some are immediate, the methods for calculating the agent requirements and agents available are different for the two types of queues. All of the agent requirements and agents available are combined into one score, however, so that the order or method of queue evaluation is irrelevant. The optimal schedule is the schedule with the best agent requirement score of all of the potential schedules analyzed. The optimal schedule is output to the user via the interface 302.

In one embodiment, the queue analysis designated by 208 in **Figure 2** and by 308 in **Figure 3** is a forward push discrete event modeler. The forward push discrete event modeler will be described with reference to **Table 1** through **Table 3**. In each interval, workload is computed by multiplying the forecast contact volume with the forecast average handling time. Capacity is computed by multiplying the number of available agents with the number of seconds in the interval in which they will work on a particular queue. If agents are capable of working on multiple queues in the same time interval, the time they spend on each of the queues is determined by static analysis or occasional explicit simulation of contact arrivals.

Referencing **Table 1**, the forward push modeler iterates over all intervals starting with the earliest interval and subtracts the capacity from the first interval's workload until all of the first interval's workload is completed. Next, the forward push modeler starts with the second earliest interval and subtracts the capacity from the second interval's workload until all of the second interval's workload is completed. This continues until all of the capacity is used or all of the workload is completed.

Table 1

Initially:						
Interval	1	2	3	4	5	6
Workload	100	100	100	0	0	0
Capacity	40	40	40	40	40	40
After the 1st Iteration:						
Interval	1	2	3	4	5	6
Workload	0	100	100	0	0	0
Capacity	0	0	20	40	40	40
After the 2nd Iteration:						
Interval	1	2	3	4	5	6
Workload	0	0	100	0	0	0
Capacity	0	0	0	0	0	40
After the 3rd. (Final) Iteration						
Interval	1	2	3	4	5	6
Workload	0	0	60	0	0	0
Capacity	0	0	0	0	0	0

Referencing **Table 2**, the forward push modeler returns an approximate percentage workload completed within that service time by evaluating the workload remaining (if any) once the number of intervals in the service goal time has elapsed. The percentage workload completed is interchangeable with the traditional measure of PCA, and will be referred to as PCA herein. In **Table 2**, the service goal time is two intervals. The average speed to answer (ASA) is computed by doing a weighted average of the amounts of workload completed in

various intervals and the time elapsed. Interval 1 in **Table 2** would have a PCA of 80% because 80 seconds of workload out of 100 seconds were completed within two intervals. In various embodiments, other performance measure than PCA can be determined.

Table 2

Service Goal Time = 2 Intervals						
Interval (Initial)	1	2	3	4	5	6
Workload	100	100	100	0	0	0
Capacity (Remaining)	40	40	40	40	40	40
Workload	0	0	60	0	0	0
Capacity	0	0	0	0	0	0
PCA	80%	20%	0%	-	-	-

Agents available and agents required are computed from the results of the forward push modeler such that the trade-offs with immediate queues that typically have explicit agent requirement can be computed and compared. Agent requirement is calculated by multiplying the workload and the required service goal percentage. Agents available is calculated by multiplying the workload and the PCA and adding the remaining capacity, of there is any. If agent requirement is greater than agents available, then the contact center is understaffed. If agents available is greater than agent requirement, then the contact center is overstaffed. Put another way, PCA exceeds required service goal percentage, or there is unused capacity. With reference to **Table 3**, the example above, the required service goal percentage is 70 and the service goal time is two intervals. The agent requirement and agents available are shown for each interval in the schedule period.

Table 3

Required Service Goal Percent = 70% Service Goal Time = 2 Intervals						
Interval (Initial)	1	2	3	4	5	6
Workload	100	100	100	0	0	0
Capacity (Remaining)	40	40	40	40	40	40
Workload	0	0	60	0	0	0
Capacity	0	0	0	0	0	0
PCA	80%	20%	0%	-	-	-
Required	70	70	70	0	0	0
Available	80	20	0	-	-	-

Figure 4 is a more detailed flow diagram showing the generation of an agent requirement score for deferred queues. The deferred queue analysis 408 is a forward push discrete event modeler. The forward push discrete event modeler 408 receives a potential schedule from the search engine 406. The potential schedule includes capacity for every interval in the schedule period. The forward push discrete event modeler 408 also receives a service goal expressed as time for every interval in the schedule period and a workload for every interval in the schedule period. The forward push event modeler iterates as shown at 408 and as previously described. The agents available formula 414 receives a workload completed percentage for every interval in the schedule period and a capacity for every interval in the schedule period. The agents available formula 414 also receives an agents required figure for every interval which is generated by the agents required formula 416. The agents available formula generates an agents available figure for every interval.

The agents required formula generates the agents required figure from the workload for every interval and the service goal for every interval. The agent requirement score

formula 412 receives the agents available figure and the agents required figure and outputs an agent requirement score for the schedule period.

The invention has been described with reference to specific embodiments and examples. The scope of the invention is defined by the claims, and includes modifications that may be made by one of ordinary skill in the art.